View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Jurnal Nasional Teknik Elektro

Vol: 3 No.1 Maret 2014 ISSN: 2302-2949

TEKNIK REKTIFIKASI CITRA DAN TAPIS KALMAN DALAM MENGESTIMASI KECEPATAN KENDARAAN

Rika Favoria Gusa
Teknik Elektro, Fakultas Teknik, Universitas Bangka Belitung
Pangkal Pinang, Provinsi Kepulauan Bangka Belitung
e-mail: rika_favoria@yahoo.com

Abstrak—Estimasi kecepatan akan sulit dilakukan dengan langsung mengolah runtun citra kendaraan yang diperoleh dengan menggunakan kamera. Hal ini dikarenakan panjang maupun luas objek-objek di dalam citra mengalami perubahan akibat adanya proyeksi perspektif.Dalam penelitian ini, distorsi geometrik objek di dalam runtun citra akan diperbaiki dengan melakukan rektifikasi citra. Selanjutnya, algoritma tapis Kalman dijalankan guna mengolah informasi dari *rectified images* yang merupakan hasil rektifikasi runtun citra sehingga kecepatan kendaraan yang diamati dapat diperkirakan.Distorsi geometrik objek di dalam runtun citra yang diakibatkan oleh proyeksi perspektif pada kamera dapat dikoreksi dengan baik menggunakan rektifikasi citra yang dilakukan dengan menerapkan matriks transformasi proyektif yang dihitung berdasarkan kondisi (panjang maupun besar sudut) sebenarnya dari garis-garis objek di dalam runtun citra.Galat estimasi kecepatan rata-rata kendaraan ialah sebesar ±3 km/jam (saat kendaraan bergerak lurus).

Kata Kunci : Estimasi, Kecepatan kenderaan, Rektifikasi citra dan Filter Kalman

Abstract— Estimating is a challenging task when the image sequence from a camera are directly processed because there is perspective projection that causes length and area ratio of objects in the image are not preserved. In this paper, it was used image rectification technique and Kalman filter algorithm to overcome the problems encountered in order to obtain accurate vehicle velocity estimation. Rectified images as result of image rectification were processed, then Kalman filter algorithm was executed based on the processing result of the rectified images. The result of the tests showed that geometric distortion on the objects in the image sequence could be corrected well by using image rectification. Kalman filter algorithm was also good enough in estimating vehicle velocity. The error of average velocity estimation was ±3 km/hour.

Keywords : Estimation, Vehicle velocity, Image rectification, Kalman filter.

I. PENDAHULUAN

Kemajuan ilmu pengetahuan dan teknologi menjadikan sebuah kamera dapat digunakan untuk memperkirakan kecepatan kendaraan.Estimasi kecepatan akan sulit dilakukan dengan langsung mengolah runtun citra kendaraan yang diperoleh dengan menggunakan kamera. Hal ini dikarenakan panjang maupun luas objek-objek di dalam citra mengalami perubahan akibat adanya proyeksi perspektif [4].Untuk itu, terlebih dahulu perlu dilakukankoreksi terhadap distorsi geometrik objek di dalam citra sehingga estimasi kecepatan kendaraan dapat dilakukan dengan mudah.

Banyak penelitian yang telah dilakukan untuk mengestimasi kecepatan kendaraan.Rezaei dan Sengupta[7] melakukan estimasi posisi, arah dan kecepatan dari sebuah mobil dengan menggunakan sistem DGPS-VS (*Differential Global Positioning System - Vehicle Sensor*) yang diintegrasikan dengan tapis Kalman.Penelitian lainnya dilakukan oleh Marslindkk. [6]yang mengestimasi kecepatan dan arah kendaraan melalui pemodelan berupa representasi geometris 3-D dari kendaraan yang diamati. Grammatikopoulosdkk.[5]melakukan estimasi kecepatan kendaraan melalui proses rektifikasi citra dan penentuan faktor skala antara bidang citra dan bidang *ground*. Estimasi kecepatan kendaraan juga dilakukan oleh Cathey dan Dailey[2] dengan menggunakan teknik *straightening* untuk menghilangkan efek perspektif dalam runtun citra dan teknik

korelasi untuk memperoleh faktor skala yang dibutuhkan.

Penelitian yang dilakukan penulis menggabungkan beberapa metode dalam penelitian-penelitian yang telah disebutkan sebelumnya.Dalam penelitian ini, distorsi geometrik objek di dalam runtun citra akan diperbaiki dengan melakukan rektifikasi citra. Selanjutnya, algoritma tapis Kalmandijalankan guna mengolah informasi dari *rectified images*yang merupakan hasil rektifikasi runtun citra sehingga kecepatan kendaraan yang diamati dapat diperkirakan.

II. METODE PENELITIAN

Penelitian dimulai dengan melakukan akuisisi data berupa video rekaman kendaraan bergerak yang diambil daritempatdengan ketinggian ±8 meter dalam format AVI. Terlebih dahulu ditentukan syarat atau kondisi yang harus dipenuhi dalam akuisisi video agar tahapan selanjutnya dapat dilakukan dengan mudah. Syarat-syarat tersebut ialah terdapat batas jalan atau garis pemisah lajur pada jalan yang dilalui kendaraan yang diamati, jalan tidak menanjak, lalu lintas kendaraan di jalan tersebut tidak padat, kamera untuk merekam gerak kendaraan dijaga statis (tidak mengikuti gerak kendaraan), tidak ada halangan pandangan di sekitar kamera serta *vanishing point* berada jauh di luar citra.Agar komputasi dapat dilakukan dengan lebih cepat, hanya dipilih *frame-frame* video dengan interval 5 *frame*.Untuk seterusnya, *frame-frame* yang dipilih disebut dengan runtun citra dan disimpan dalam format jpeg dengan ukuran 480x640.

Proses-proses yang dikenakan pada runtun citra adalah sebagai berikut:

I. Rektifikasi citra

Rektifikasi citra dilakukan dengan menghitung matriks transformasi proyektif yang terdiri dari matriks transformasi *pure projective,* matriks transformasi *affine* dan matriks transformasi *similarity.*Ketiga matriks diterapkan pada runtun citra untuk menghilangkan distorsi geometrik objek di dalam citra.

a. Transformasi *pure projective*

1. Menghitung citra latar belakang (*background image*)

Untuk memperoleh matriks transformasi *pure projective*, terlebih dahulu dihitung citra latar belakang dari runtun citra. Citra latar belakang diperoleh dengan melakukan *averaging* yaitu menjumlahkan nilai intensitas semua *frame* dalam runtun citra lalu membaginya dengan jumlah *frame* yang ada. Citra latar belakang yang dihasilkan hanya akan mengandung objek-objek statis (tidak bergerak).

2. Mendeteksi garis lurus di dalam citra latar belakang

Garis lurus di dalam citra latar belakang diperlukan untuk menghitung *vanishing point.*Deteksi garis lurus dilakukan dengan menggunakan transformasi Hough.Untuk dapat menggunakan transformasi Hough, citra latar belakang yang merupakan citra RGB harus diubah menjadi citra keabuan (*grayscale image*) terlebih dahulu.Kemudian, dilakukan deteksi tepi (*edge detection*) pada citra keabuan dengan metode Canny.

Tiap koordinat titik (*x,y*) yang menyusun tepi objek di dalam citra latar belakang dinyatakan dalam parameter ρ dan θ karena transformasi Hough merepresentasikan setiap garis lurus di dalam citra melalui persamaan (1).

$$\rho = x.\cos\theta + y.\sin\theta$$
$$\text{dengan } -90 < \theta \leq 90 \qquad (1)$$

Melalui mekanisme voting, dapat diketahui jumlah titik untuk tiap pasangan nilai ρ dan θ. Titik-titik dengan nilai ρ dan θ yang sama akan membentuk sebuah garis lurus.

3. Menghitung koordinat *vanishing points*

Untuk menghitung koordinat *vanishing point,* dipilih dua garis yang (sebenarnya) sejajar dari garis-garis di dalam citra latar belakang yang dideteksi dengan menggunakan transformasi Hough.Jika deteksi dengan transformasi Hough tidak menghasilkan garis-garis yang (sebenarnya) sejajar, maka digunakan garis-garis sejajar yang dimiliki objek di dalam citra latar belakang.Dalam penelitian ini, garis lurus yang digunakan ialah garis-garis batas jalan yang vertikal (diasumsikan sebenarnya sejajar) di dalam citra latar belakang.

Persamaan garis-garis sejajar diperoleh dengan melakukan *cross product* terhadap koordinat titik-titik ujung garis. Koordinat titik-titik ujung garis (*x,y*) harus diubah ke bentuk koordinat homogen[x y w] dengan w = 1 terlebih dahulu. Kemudian, koordinat *vanishing*

point dihitung menggunakan *cross product* dari koefisien-koefisien persamaan garis-garis sejajar.Sedikitnya diperlukan dua *vanishing point* untuk menghitung *vanishing line.*Dikarenakan bidang citra diasumsikan sejajar terhadap sumbu x bidang *world*, maka *vanishing point* 2 berada di tak hingga sumbu x citra..

4. Menghitung persamaan *vanishing line*

Vanishing line dihitung menggunakan *cross product* koordinat kedua *vanishing point* dalam bentuk koordinat homogen.*Vanishing line* direpresentasikan dalam bentuk $[l_1 \quad l_2 \quad l_3]$ dengan l_1, $l_2$dan l_3 merupakan koefisien-koefisien persamaan *vanishing line* yang diperoleh.

5. Menerapkan matriks transformasi *pure projective* ke runtun citra

Setelah persamaan *vanishing line* diketahui, maka matriks transformasi *pure projective* (Pp) dapat dihitungdengan menggunakan persamaan (2).

$$Pp = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \qquad (2)$$

Koordinat titik di citra yang "baru" diperoleh dengan mengalikan matriks transformasi *pure projective* (Pp) dengan koordinat titik di dalam citra asal. Runtun citra "baru" yang dihasilkan disebut *affine-rectified images.* Dengan diterapkannya matriks transformasi *pure projective,* keparalelan garis dan perbandingan luas objek pada citra dapat dikembalikan sesuai dengan keadaan yang sebenarnya.

b. Transformasi *affine*

1. Menghitung citra latar belakang

Citra latar belakang dihitung kembali dari *affine-rectified images* dengan melakukan *averaging,* sama dengan cara yang dilakukan saat menghitung citra latar belakang sebelumnya.

2. Menghitung titik pusat dan jari-jari lingkaran *constraint*

Ada tiga jenis *constraint* yang dapat dipilih sesuai dengan informasi yang tersedia mengenai keadaan sebenarnya dari objek-objek yang terdapat di dalam citra latar belakang, yaitu besar sudut sebenarnya antara dua garis (*known angle*), dua sudut yang besarnya sama yang diapit dua pasang garis dengan arah berbeda (*equal angles*) dan perbandingan panjang sebenarnya antara dua garis tidak sejajar (*known length* ratio) [3].

Dalam penelitian ini, *constraint* yang dipilih ialah *known angle* dan *known length ratio* karena dapat diketahui besar sudut sebenarnya dan perbandingan panjang sebenarnya dari garis-garis objek di dalam citra latar belakang.Dilakukan pengukuran langsung di lokasi perekaman video terhadap panjang sebenarnya dari garis batas parkir kendaraan (baik garis horizontal maupun garis vertikal) dan besar sudut sebenarnya pada kanopi (tempat tanaman rambat).Dengan mengetahui koordinat titik-titik ujung garis-garis di dalam citra latar belakang baik yang mengapit sudut dengan besar diketahui (tempat tanaman rambat) maupun yang perbandingan panjangnya diketahui (garis batas parkir kendaraan), dapat dihitung titik pusat dan jari-jari lingkaran untuk tiap *constraint* yang dipilih.

Untuk *known angle constraint,* terlebih dahulu dihitung persamaan kedua garis yang mengapit sudut (θ) dengan besar diketahui.Persamaan garis diperoleh dengan melakukan *cross product* terhadap koordinat titik-titik ujung garis yang telah diubah ke bentuk koordinat homogen.Garis pertama (disebut l_a) dan garis kedua (disebut l_b)dinyatakan dalam vektor yang terdiri dari tiga elemen yaitu $l_a = [l_a(1), l_a(2), l_a(3)]$ dan $l_b = [l_b(1), l_b(2), l_b(3)]$. Elemen-elemen vektor merupakan koefisien-koefisien persamaan masing-masing garis. Selanjutnya, dihitung arah kedua garis melalui persamaan (3),

$a = -l_a(2) / l_a(1)$dan
$b = -l_b(2) / l_b(1)$ (3)

sehingga diperoleh lingkaran *constraint* dengan titik pusat

$$(c_\alpha, c_\beta) = \left(\frac{(a+b)}{2}, \frac{(a-b)}{2} cot\theta\right) \qquad (4)$$

dan jari-jari

$$r = \left|\frac{(a-b)}{2\ sin\theta}\right| \qquad (5)$$

Constraint ke-2 yang dipilih ialah *known length ratio.*Dengan mengetahui koordinat titik-titik ujung garis-garis di dalam citra latar belakang yang perbandingan panjang sebenarnya diketahui, dapat dihitung Δx_ndan Δy_n yang diperlukan untuk memperoleh koordinat titik pusat dan jari-jari lingkaran

*constraint.*Titik pusat dan jari-jari lingkaran *constraint* dihitung berdasarkan persamaan (6) dan (7).

Titik pusat lingkaran

$$(c_\alpha, c_\beta) = (\frac{\Delta x_1 \Delta y_1 - s^2 \Delta x_2 \Delta y_2}{\Delta y_1^2 - s^2 \Delta y_2^2}, 0) \quad (6)$$

Jari-jari lingkaran

$$r = \left|\frac{s(\Delta x_2 \Delta y_1 - \Delta x_1 \Delta y_2)}{\Delta y_1^2 - s^2 \Delta y_2^2}\right| \quad (7)$$

3. Memperoleh parameter α dan β

Setelah menghitung titik pusat dan jari-jari lingkaran dari tiap *constraint* yang dipilih, lingkaran-lingkaran *constraint* tersebut kemudian direpresentasikan pada bidang kompleks dengan α dan β sebagai komponen riil dan imajinernya.Untuk memperoleh nilai parameter α dan β yang tepat guna memperoleh matriks transformasi *affine*, dihitung titik potong dari lingkaran-lingkaran *constraint* yang diperoleh.

4. Menerapkan matriks transformasi *affine* ke *affine-rectified images*

Setelah memperoleh nilai parameter α dan β, matriks transformasi *affine* dihitung dengan menggunakan persamaan (8).

$$\text{Aff} = \begin{bmatrix} \frac{1}{\beta} & \frac{-\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Runtun citra yang telah dikenakan matriks transformasi *affine* disebut *metric-rectified images.* Matriks transformasi *affine* mengembalikan besar sudut dan perbandingan panjang dari garis-garis tidak sejajar di dalam citra sehingga sesuai dengan keadaan yang sebenarnya.

c. Transformasi *similarity*

Untuk memperoleh citra yang menggambarkan keadaan sesuai dengan keadaan sebenarnya pada saat perekaman video, perlu dilakukan rotasi citra.Sudut rotasi citra (θ) dihitung dari garis-garis objek di dalam citra.Dalam penelitian ini, dihitung sudut rotasi yang diperlukan agar garis batas jalan yang masih miring di dalam *metric-rectified images* menjadi tegak lurus sesuai dengan keadaan pada saat perekaman video. Dilakukan rotasi citra searah jarum jam untuk memperoleh citra yang dapatmenggambarkan kondisi yang sebenarnya.

Selain besar sudut rotasi citra (θ), nilai *isotropic scaling* citra (s) serta translasi citra pada arah sumbu x (t_x) dan arah sumbu y (t_y) juga perlu ditentukan. Dalam penelitian ini, tidak dilakukan perubahan ukuran citra maupun penggeseran titik pusat citra sehingga *isotropic scaling* (s) = 1 dan translasi citra (t_x dan t_y) = 0.

Selanjutnya, matriks transformasi *similarity* dihitung menggunakan persamaan (9).

$$\text{Sim} = \begin{bmatrix} s.cos\theta & -s.sin\theta & t_x \\ s.sin\theta & s.cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Runtun citra yang telah dikenakan matriks transformasi *similarity* disebut *rectified images.* Keadaan objek-objek di dalam*rectified images* telah sesuai dengan keadaan sebenarnya pada bidang *world.*

II. Penghitungan faktor skala

Langkah awal yang dilakukan dalam tahapan ini ialah menghitung citra latar belakang dari *rectified images* yang telah diperoleh. Citra latar belakang dihitung dengan melakukan *averaging,* sama dengan cara yang dilakukan saat menghitung citra latar belakang sebelumnya.

Untuk memperoleh faktor skala yang menunjukkan perbandingan antara ukuran objek di dalam citra (dalam satuan piksel) dengan ukuran objek sebenarnya (dalam satuan meter), digunakan salahsatu objek di dalam citra latar belakang yang ukuran sebenarnya diketahui.Telah dilakukan pengukuran langsung di lokasi perekaman video terhadap panjang sebenarnya dari garis batas parkir kendaraan baik arah horizontal maupun vertikal. Dengan membandingkan panjang sebenarnya dengan panjang garis tersebut di dalam citra, maka dapat diperoleh faktor skala yang akan digunakan dalam algoritma tapis Kalman.

III. Penghitungan *centroid* kendaraan yang diamati

Untuk dapat menghitung *centroid* kendaraan yang diamati, perlu dilakukan *background subtraction* terlebih dahulu.*Background subtraction* dilakukan untuk memperoleh citra latar depan (*foreground image*) dengan cara mengurangkan citra (dalam hal ini ialah *rectified image*)

dengan citra latar belakangnya (*background image*). Citra latar depan yang dihasilkan hanya mengandung objek-objek bergerak. Selanjutnya, citra latar depan yang merupakan citra RGB diubah menjadi citra biner yang terdiri dari objek berwarna putih dan latar berwarna hitam.

Di dalam citra biner, sebuah objek dapat dideteksi menjadi dua atau lebih objek.Hal ini terjadi karena nilai intensitas pada bagian tertentu objek berada di bawah nilai *threshold* saat binerisasi sehingga bagian objek tersebut menjadi latar di dalam citra biner.Akibatnya, objek yang semula berjumlah satu dapat terpisah menjadi dua objek atau lebih.Untuk mengatasi hal tersebut, dilakukan operasi morfologi baik dilasi maupun erosi pada citra biner.Operasi morfologi juga digunakan untuk mengurangi derau berupa objek-objek lain yang tidak diinginkan di dalam citra.

Untuk melakukan operasi dilasi maupun erosi, dibutuhkan unsur penstruktur (*structuring element*) yang tepat.Dalam penelitian ini, digunakan unsur penstruktur berupa persegi panjang (*rectangle*) karena objek yang diamati adalah kendaraan.Setelah bagian-bagian kendaraan di dalam citra menyatu, dapat dilakukan penghitungan *centroid.*Selain kendaraan yang diamati, juga terdapat kendaraan lain maupun benda bergerak lainnya (misal daun yang bergerak) di dalam citra biner yang tidak benar-benar hilang setelah operasi morfologi dilakukan. Oleh karena itu, dilakukan pelabelan terhadap semua objek yang terdeteksi di dalam citra biner.

Selanjutnya, dihitung nilai rata-rata (*mean*) baris dan kolom masing-masing objek.Nilai rata-rata inilah yang menjadi koordinat *centroid* tiap objek yang terdapat di dalam citra biner sehingga koordinat *centroid* kendaraan yang diamati yang merupakan salahsatu objek di dalam citra biner dapat diketahui. Dengan melakukan prosedur yang sama seperti di atas ke semua citra (*rectified images*), maka dapat diperoleh koordinat *centroid* kendaraan yang diamati di dalam runtun citra.

IV. Penggunaan algoritma tapis Kalman

Untuk dapat menggunakan tapis Kalman, proses yang diukur harus dimodelkan dengan persamaan linear terlebih dahulu. Dalam penelitian ini, proses yang diukur ialah kendaraan yang bergerak di jalan dan *state variable* proses yang ingin diperoleh nilainya ialah posisi dan kecepatan kendaraan. Dikarenakan citra merupakan bidang dua dimensi, maka posisi dan kecepatan kendaraan dihitung untuk masing-masing arah (sumbu x dan sumbu y citra).*State vector* (**x**) yang digunakan dalam pemodelan adalah

$$\mathbf{x}=\begin{bmatrix} x \\ y \\ V_x \\ V_y \end{bmatrix}$$

dengan

x dan y :posisi kendaraan

V_x :kecepatan kendaraan arah sumbu x citra

V_y :kecepatan kendaraan arah sumbu y citra

Tidak ada input yang diberikan pada proses (u = 0) karena yang dilakukan pada penelitian ini hanya pengamatan terhadap kendaraan yang bergerak.

Dalam penelitian ini, persamaan linear yang merupakan hukum fisika dasar yang digunakan untuk memperoleh bentuk pemodelan adalah sebagai berikut:

$$x_k = x_{k-1} + \mathrm{dt}.(Vx)_{k-1}$$
$$y_k = y_{k-1} + \mathrm{dt}.(Vy)_{k-1}$$
$$(Vx)_k = (Vx)_{k-1}$$
$$(Vy)_k = (Vy)_{k-1}$$

dengan dt ialah waktu yang dibutuhkan kendaraan untuk bergerak dari posisi x_{k-1} ke x_k atau dari posisi y_{k-1} ke y_k (interval waktu antar *frame* dalam runtun citra). k menunjukkan langkah waktu (*time step*).

Pemodelan dilakukan dengan menggunakan persamaan (10) dan (11).

Model keadaan

$$\mathbf{x}_k = \mathrm{A}.\mathbf{x}_{k-1} + \mathrm{B}.u_{k-1} + w_{k-1} \quad (10)$$

Model pengukuran

$$z_k = \mathrm{C}.\mathbf{x}_k + v_k \quad (11)$$

Berdasarkan empat persamaan linear di atas dan hasil pengukuran proses (z) berupa posisi (koordinat *centroid*) kendaraan di dalam citra, maka pemodelan yang diperoleh adalah sebagai berikut:

Model keadaan

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + w_{k-1}$$

Model pengukuran

$$z_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_k + v_k$$

dengan

w: vektor derau proses dengan kovarians Q

v: vektor derau pengukuran dengan kovarians R

Kovarians Q dan R harus dihitung terlebih dahulu agar algoritma tapis Kalman dapat dijalankan. Dilakukan juga inisialisasi untuk dapat menjalankan algoritma tapis Kalman yaitu dengan menentukan nilai awal estimasi *state* proses dan kovarians galatnya. Faktor skala digunakan dalam penentuan nilai awal estimasi *state* proses agar posisi dan kecepatan kendaraan yang dihasilkan algoritma tapis Kalman mempunyai satuan meter dan meter per detik.Dihitung juga galat estimasi posisi dan kecepatan kendaraan. Galat estimasi posisi kendaraan dihitung dengan membandingkan posisi kendaraan yang diperoleh dari proses pengolahan runtun citra (*centroid* kendaraan) dengan hasil estimasi posisi kendaraan menggunakan algoritma tapis Kalman. Galat estimasi kecepatan kendaraan dihitung dengan membandingkan kecepatan kendaraan yang sebenarnya dengan hasil estimasi kecepatan menggunakan algoritma tapis Kalman.

III. HASIL DAN PEMBAHASAN

3.1. Rektifikasi Citra

a. Transformasi *pure projective*

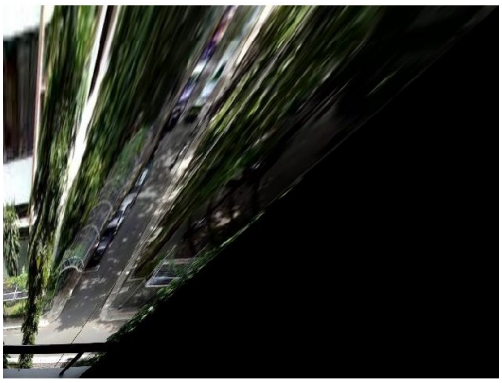
Hal pertama yang harus dilakukan untuk dapat memperoleh matriks transformasi *pure projective* ialah menghitung citra latar belakang dari runtun citra yang diperoleh.Selanjutnya, dideteksi garis-garis lurus yang ada di dalam citra latar belakang untuk menghitung koordinat *vanishing point.*Citra latar belakang (disajikan dalam citra keabuan) dan garis-garis yang digunakan untuk menghitung koordinat *vanishing point*dapat dilihat dalam Gambar 1.

Gambar 1 Garis-garis yang digunakan untuk menghitung koordinat *vanishing point*

Diperoleh koordinat *vanishing point* 1 yaitu [307, -85, 1] dalam bentuk koordinat homogenatau (307,-85) dalam bentuk koordinat kartesian.*Vanishing point* ini berada di luar citra.sedangkankoordinat *vanishing point* 2 (dalam bentuk homogen) ialah [7980, 0, 0] yang berarti bahwa *vanishing point* 2 berada di tak hingga sumbu x citra.Setelah koordinat kedua *vanishing point* diketahui, dihitung persamaan *vanishing line.* Diperoleh persamaan *vanishing line* yaitu 0,0118y + 1 = 0 atau y = -84,75 yang berarti garis ini berada di luar citra sehingga dihasilkan matriks transformasi *pure projective* sebagai berikut:

$$Pp = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0{,}0118 & 1 \end{bmatrix}$$

Matriks transformasi *pure projective* diterapkan ke runtun citra untuk mengembalikan keparalelan garis dan perbandingan luas objek pada citra sehingga sesuai dengan keadaan yang sebenarnya.Citra yang telah dikenakan transformasi *pure projective* disebut *affine-rectified image.*Salahsatu citra tersebut dapat dilihat pada Gambar 2.



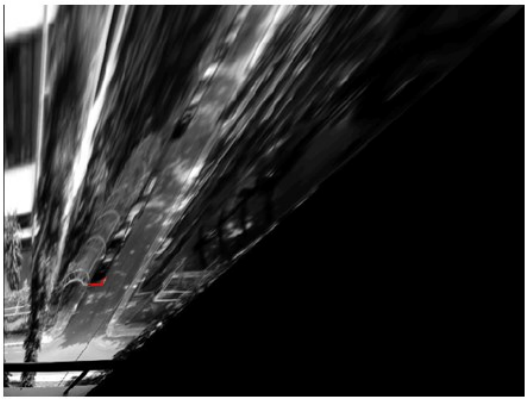
Gambar 2 Salah satu a*ffine-rectified image*

Pada Gambar 2 dapat dilihat bahwa jalan yang semula menyempit menjadi paralel sesuai dengan keadaan yang sebenarnya pada bidang *world*.

b. Transformasi *affine*

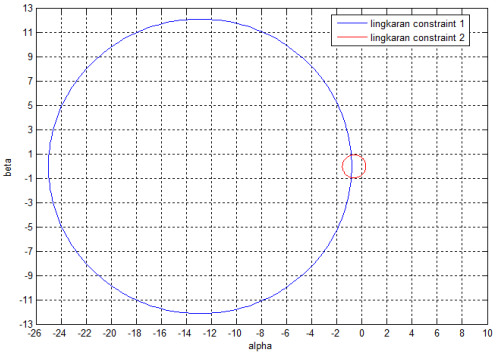
Langkah awal yang dilakukan untuk memperoleh matriks transformasi *affine* ialah menentukan *constraint* berdasarkan informasi yang tersedia di dalam citra. Citra yang digunakan untuk memilih *constraint* ialah citra latar belakang yang dihitung dari *affine-rectified images* dengan melakukan *averaging*. Dari citra latar belakang ini, dipilih dua *constraint* yaitu besar sudut sebenarnya yang dibentuk oleh dua garis (*known angle*)dan perbandingan panjang sebenarnya antara dua garis tidak sejajar (*known length ratio*) yang digunakan untuk menghitung titik pusat dan jari-jari lingkaran *constraint*.

Constraint pertama yaitu *known angle* dipilih karena diketahui besar sudut sebenarnya (pada bidang *world*)antara dua garis pada tempat tanaman merambat yang terdapat di dalam citra latar belakang. Besar sudut yang dibentuk oleh kedua garis ialah 90^{o}. Selanjutnya, c*onstraint* ke-2 yang dipilih ialah *known length ratio*. *Constraint* ini dipilih karena telah diukur panjang sebenarnya dari dua garis batas parkir mobil yang berada paling bawah dalam citra latar belakang. Rasio panjang garis horizontal terhadap panjang garis vertikal adalah 1:0,5 dengan ukuran garis sebenarnya sepanjang 1,95 meter.Garis-garis yang telah disebutkan dapat dilihat pada Gambar 3.



Gambar 3 Garis-garis yang digunakan untuk menghitung titik pusat dan jari-jari lingkaran *constraint*

Diperoleh lingkaran *constraint* 1 berpusat di (-12,91, 0) dan memiliki jari-jari sebesar 12,09 sedangkan lingkaran *constraint* 2 berpusat di (-0,625, 0) dan berjari-jari 0,9375.Lingkaran-lingkaran *constraint* direpresentasikan pada bidang kompleks dengan α dan β sebagai komponen riil dan imajiner. Nilai parameter α dan β yang tepat digunakan untuk menghitung matriks transformasi *affine* adalah titik potong dari lingkaran-lingkaran *constraint*. Kedua lingkaran *constraint* yang diperoleh dapat dilihat pada Gambar 4.

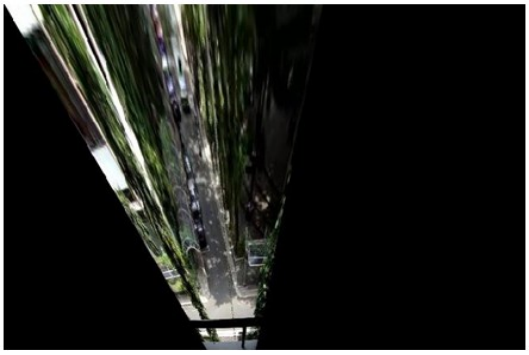


Gambar 4 Lingkaran-lingkaran *constraint*

Dari Gambar 4, dapat dilihat bahwa kedua lingkaran *constraint* berpotongan di titik (-0,8524± i. 0,9095) sehingga diperoleh nilai α=-0,8524 danβ = 0,9095.Setelah memperoleh nilai parameter α dan β, maka matriks transformasi *affine*dapat dihitung sebagai berikut:

$$\text{Aff} = \begin{bmatrix} \frac{1}{\beta} & \frac{-\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1{,}0995 & 0{,}9373 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matriks transformasi *affine* akan mengembalikan sudut dan perbandingan panjang dari garis-garis tidak sejajar di dalam citra sehingga sesuai dengan keadaan yang sebenarnya pada bidang *world*. Citra-citra yang telah dikenakan transformasi *affine* disebut *metric-rectified images.*Salahsatu dari citra tersebut dapat dilihat pada Gambar 5.



Gambar 5 Salah satu *metric-rectified image*

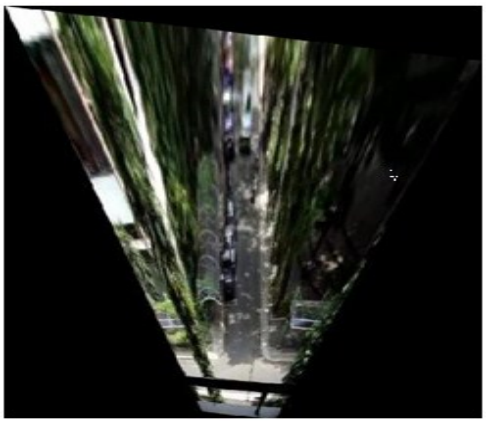
Dalam Gambar 5, terlihat bahwa besar sudut dan perbandingan panjang antara garis-garis di dalam citra yang dipilih sebagai *constraint* telah dikembalikan sesuai dengan keadaan yang sebenarnya pada bidang *world*.

c. Transformasi *similarity*

Dari *metric-rectitfiedimage* yang diperoleh, dapat dilihat bahwa masih diperlukan rotasi agar citra yang dihasilkan sesuai dengan gambaran keadaan sebenarnya saat perolehan video rekaman. Sudut rotasi citra (θ) yang diperlukan adalah sebesar 0,1806 radian sehingga diperoleh matriks transformasi *similarity* sebagai berikut:

$$\text{Sim} = \begin{bmatrix} 0{,}9837 & -0{,}1796 & 0 \\ 0{,}1796 & 0{,}9837 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Citra-citra yang telah dikenakan transformasi *similarity* disebut *rectified images. Rectified images* ini akan digunakan dalam tahapan-tahapan proses selanjutnya. Salahsatu *rectified image* dapat dilihat pada Gambar 6.



Gambar 6 Salah satu *rectified image*

Dari Gambar 6, dapat dilihat bahwa setelah melakukan tahap-tahap rektifikasi citra, distorsi geometrik pada objek-objek di dalam runtun citra asal dapat dihilangkan sehingga keadaan objek-objek tersebut sesuai dengan keadaan yang sebenarnya (pada bidang *world*).Dengan demikian, selanjutnya dengan mudah dapat dihitung koordinat *centroid* kendaraan serta faktor skala yang menunjukkan perbandingan antara ukuran objek di dalam citra (dalam satuan piksel) dengan ukuran objek sebenarnya (dalam satuan meter).

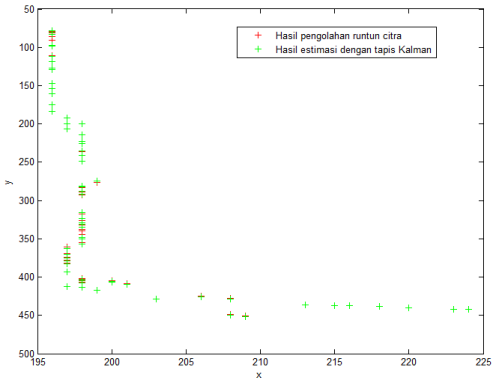
3.2. Algoritma Tapis Kalman

Algoritma tapis Kalman dimulai dengan memberikan nilai awal estimasi *state* dan kovarians galat estimasinya. Dimisalkan, nilai awal estimasi *state*$\hat{\mathrm{x}}_0$ dan kovarians galat estimasi P_0adalah sebagai berikut:

$\hat{\mathrm{x}}_0 = [(1 * 0{,}2) \quad (1 * 0{,}16) \quad 0 \quad 8]'$dan

$$P_0 = 10 * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

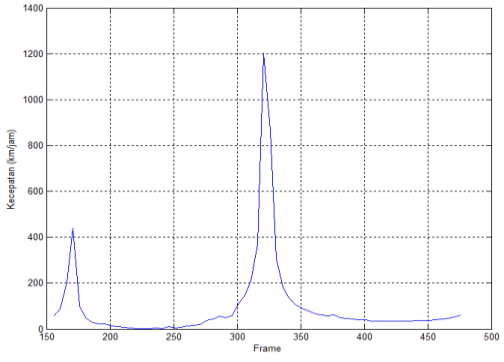
Di dalam $\hat{\mathrm{x}}_0$, posisi (x dan y) dinyatakan dalam satuan meter, kecepatan (V_x dan V_y) dinyatakan dalam satuan meter/detik. Bilangan pengali 0,2 dan 0,16 pada elemen 1 dan 2 di dalam$\hat{\mathrm{x}}_0$ merupakan faktor skala yang telah diperoleh.Setelah inisialisasi dilakukan, algoritma tapis Kalman dijalankan untuk mengestimasi posisi dan kecepatan kendaraan pada semua*frame* yang dipilih.



Gambar 7 Perbandingan koordinat *centroid* kendaraan hasil pengolahan runtun citra dengan hasil estimasi dengan tapis Kalman

Dalam penelitian ini, kendaraan yang diamati bergerak lurus searah sumbu y positif citra kemudian berbelok ke kanan (arah sumbu x positif citra).Dari Gambar 7, dapat diketahui bahwa saat kendaraanbergerak lurus, koordinat *centroid* kendaraan sebagai hasil estimasi algoritma tapis Kalman mulai mendekati koordinat*centroid* kendaraan yang merupakan hasil pengolahan runtun citra pada iterasi ke-5. Galat posisi paling tinggi terjadi pada iterasi ke-4 yaitu sebesar 7 piksel atau ±1 m.

Saat terjadi perubahan lintasan (belokan), kedua *centroid* kendaraan menjauh sehinggagalat posisinya meningkat hingga 17 piksel atau ±2,7 m pada iterasi ke-34 yang merupakan nilai galat terbesar. Namun, iterasi yang cukup membuat kedua *centroid* kendaraan dapat mendekat kembali sehingga nilai galat posisinya mendekati 0.Nilai galat yang telah disebutkan terjadi pada arah sumbu y citra.



Gambar 8 Hasil estimasi kecepatan kendaraan

Kecepatan sebenarnya dari kendaraan diketahui sebesar ±30 km/jam saat kendaraan bergerak lurus kemudian perlahan-lahan menurun menjadi ±20 km/jam saat kendaraan membelok.Dalam Gambar 8, dapat dilihat terjadi dua kali lonjakan kecepatan kendaraan. Hal ini disebabkan karena hasil estimasi posisi kendaraan menggunakan tapis Kalman pada saat tersebut mempunyai galat yang cukup besar sehingga mempengaruhi nilai estimasi kecepatan yang dihasilkan. Oleh karena itu, untuk perhitungan nilai kecepatan rata-rata kendaraan, hanya diperhitungkan iterasi-iterasi dengan nilai estimasi kecepatan yang nilai estimasi posisinya kurang dari 0,4 meter.

Dari hasil estimasi dengan tapis Kalman, diperoleh kecepatan rata-rata kendaraan saat bergerak lurus sebesar ±27 km/jam (dihitung dari nilai kecepatan pada iterasi ke-5 hingga iterasi ke-30) dan kecepatan rata-rata kendaraan saat membelok sebesar ±57 km/jam (dihitung dari nilai kecepatan pada iterasi ke-37 hingga iterasi ke-65).Dari nilai kecepatan rata-rata ini, diperoleh galat kecepatan kendaraan sebesar ±3 km/jam saat bergerak lurus dan ±37 km/jam saat membelok.

Hal ini menunjukkan bahwa algoritma tapis Kalman belum dapat memberikan estimasi kecepatan kendaraan yang cukup baik pada saat kendaraan membelok (berubah lintasan). Penyebabnya adalah *centroid* kendaraan sebagai hasil pengolahan runtun citra yang digunakan sebagai masukan/pengukuran bagi algorima tapis Kalman kurang tepat dalam menunjukkan posisi *centroid* kendaraan yang sebenarnya pada saat kendaraan tersebut membelok.

IV. KESIMPULAN

1. Distorsi geometrik objek di dalam runtun citra yang diakibatkan oleh proyeksi perspektif pada kamera dapat dikoreksi dengan baik menggunakan rektifikasi citra yang dilakukan dengan menerapkan matriks transformasi proyektif yang dihitung berdasarkan kondisi (panjang maupun besar sudut) sebenarnya dari garis-garis objek di dalam runtun citra.
2. Galat estimasi kecepatan rata-rata kendaraan ialah sebesar ±3 km/jam (saat kendaraan bergerak lurus).
3. Algoritma tapis Kalman tidak menghasilkan estimasi kecepatan kendaraan yang cukup

baik pada saat kendaraan membelok dikarenakan koordinat *centroid* kendaraan yang digunakan sebagai masukan/pengukuran bagi algorima tapis Kalman kurang tepat dalam menunjukkan posisi *centroid* kendaraan yang sebenarnya pada saat kendaraan tersebut membelok.

DAFTAR PUSTAKA

[1] Brown, R. G., Hwang, P. Y. C. Introduction to Random Signals and Applied Kalman Filtering. 3rd ed.Canada: John Wiley and Sons.1997.

[2] Cathey, F. W., Dailey, D. J. *A Novel Technique to Dynamically Measure Vehicle Speed using Uncalibrated Roadway Cameras.* Proceedings of IEEE.2005.

[3] Liebowitz, D., Zisserman, A.*Metric Rectification for Perspective Images of Planes*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.1998.

[4] Maduro, C., Batista, K., Peixoto, P., Batista, J.*Estimation of Vehicle Velocity and Traffic Intensity using Rectified Images*. Proceedings of the 2008 IEEE International Conference on Image Processing.2008:777-780.

[5] Grammatikopoulos, L., Karras, G., Petsa, E.Automatic Estimation of Vehicle Speed from Uncalibrated Video Sequences.*International Archives of Photogrammetry and Remote Sensing*.2002.

[6] Marslin, R., Sullivan, G. D., Baker, K. D. Kalman Filters in Constrained Model Based Tracking. Department of Computer Science University of Reading.2008.

[7] Rezaei, S.,Sengupta, R. Kalman Filter Based Integration of DGPS and Vehicle Sensors for Localization.IEEE Transactions on Control Systems Technology.2007; 15 (6).

[8] Simon, D.Kalman Filtering. Department of Electrical and Computer Engineering Cleveland State University.2001.

[9] Gusa, R., Favoria. Estimasi Kecepatan Kendaraan Menggunakan Rektifikasi Citra dan Tapis Kalman. Tesis Pascasarjana Teknik Elektro Universitas Gadjah Mada. 2010.

Biodata Penulis

Rika Favoria Gusa, Dosen pada Program Studi Teknik Elektro Fakultas Teknik Universitas Bangka Belitung, menyelesaikan pendidikan Strata 1 (satu) di Program Studi Fisika Teknik UGM, dan Strata 2 (dua) di Program Studi Teknik Elektro UGM. Bidang keahlian yang digeluti adalah Pengolahan Citra Digital dan *Computer Visio.*